5. What command-line arguments are used for running NSE scripts?

When running NSE scripts in **Nmap**, you can use several command-line arguments to specify scripts, control behavior, and customize their execution. Here's a detailed guide:

---

1. Specifying Scripts

a. `--script`

- Used to specify one or more NSE scripts to run.
- You can provide:
  - Script names.
  - Script categories.
  - Wildcards for pattern matching.

Examples:

- Run a single script:

```
nmap --script http-title <target>
```

- Run multiple scripts (comma-separated):

```
nmap --script http-title,dns-brute <target>
```

- Run scripts by category:

```
nmap --script vuln <target>
```

- Use wildcards:

```
nmap --script "http-*" <target>
```

---

2. Providing Arguments to Scripts

a. `--script-args`

- Pass additional parameters to a script that accepts arguments.
- Format: `key=value` pairs, separated by commas.

Examples:

- Single argument:

```
nmap --script http-brute --script-args userdb=/path/to/users.txt <target>
```

- Multiple arguments:

```
nmap --script http-form-brute --script-args
"userdb=/users.txt,passdb=/passwords.txt" <target>
```

- Using nested arguments:

```
nmap --script ssl-cert --script-args "ssl-cert.timeout=5000"
```

---

3. Debugging and Verbose Output

a. `--script-trace`

- Provides detailed debugging information for the script's execution.
- Useful for troubleshooting or understanding a script's behavior.

```
nmap --script http-title --script-trace <target>
```

b. `-v` or `-vv`

- Increases verbosity to show more details about the script’s progress.

```
nmap -v --script dns-brute <target>
```

---

4. Controlling Parallelism

a. `--script-threads`

- Sets the number of threads for running NSE scripts.
- Default is **1**, but increasing this can improve performance for scripts like brute-force attacks.

Example:

```
nmap --script ftp-brute --script-threads 10 <target>
```

---

5. Limiting Scripts

a. `--script-timeout`

- Sets a maximum execution time for scripts.
- Prevents scripts from hanging indefinitely.

Example:

```
nmap --script vuln --script-timeout 30s <target>
```

---

6. Combining with Other Scans

- NSE scripts can be combined with other Nmap scan options like port scanning or version detection.

Examples:

- Combine with TCP SYN scan:

```
nmap -sS --script vuln <target>
```

- Combine with service version detection:

```
nmap -sV --script http-title <target>
```

---

7. Disabling Default Scripts

a. `--script=default`

- Runs default scripts (e.g., `banner`, `ssl-cert`).

```
nmap --script default <target>
```

b. `--script "not-default"`

- Excludes default scripts from the scan.

```
nmap --script "not default,vuln" <target>
```

---

8. Specifying Script Categories

a. Categories

- Instead of individual scripts, you can specify categories to run all related scripts.
- Categories include:
  - `auth`, `broadcast`, `brute`, `default`, `discovery`, `exploit`, `external`, `fuzzer`, `intrusive`, `malware`, `safe`, `version`, `vuln`.

Example:

Run all **vuln** category scripts:

```
nmap --script vuln <target>
```

---

9. Script Scans with Target Specification

a. Targeting Specific Ports

- Specify ports to scan:

```
nmap -p 80,443 --script http-title <target>
```

b. Targeting Multiple Hosts

- Provide multiple targets:

```
nmap --script ftp-brute <target1> <target2>
```

c. Using Input Files

- Load targets from a file:

```
nmap --script vuln -iL targets.txt
```

Summary Table of Options

Option	Description	Example
`--script`	Specify scripts to run.	`--script http-title`
`--script-args`	Provide arguments to scripts.	`--script-args userdb=/users.txt`
`--script-trace`	Enable detailed debugging output.	`--script-trace`
`--script-threads`	Control parallel execution of scripts.	`--script-threads 5`
`--script-timeout`	Limit the execution time of scripts.	`--script-timeout 30s`
`--script default`	Run default scripts.	`--script default`
`--script "not <type>"`	Exclude specific categories or scripts.	`--script "not default,vuln"`